Módulos Didáticos

DEEP LEARNING 05

LINKED EDUCATION

V1020

Regularização

Revisão

Falando um pouco mais de Bias e Variance, bagging, **dropout, L1 e L2**

Regularização

Revisão

Mais sobre Dropout:

Ocorre apenas nos treinos

Depois do dropout os pesos são minimizados na proporção do dropout (por que?)

Podemos aplicar dropout não só em camadas de NN densas, mas tb em camadas de convolução e até no input.

Cuidado: em algumas implementações, a taxa é para os deixados sem dropout.

Artigo (anexo) que deu origem à técnica: Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Regularização

L1 e L2:

Como L2 penaliza o quadrado dos pesos, quando os pesos ficam menores que um, não há tendência de “empurrá-los” a zero. Isso acontece com L1 (gera “esparsidade”).

L2 também é conhecida como weight decay.

Regularização

Melhorando o exemplo cifar10:

cnn_cifar10_res2.ipynb

Transfer Learning

Revisão

Vamos usar o aprendizado da rede MobileNet V2 e adaptá-la para separar X de O.

Base: transfer_MobileNet.ipynb

Parte 1: criar um pequeno dataset de X e O e rotulá-lo (colocando em pastas diferentes)

Parte 2: Eliminar a camada output da MobileNet V2 , colocando o output da penúltima camada como input no novo modelo (modelox)

Transfer Learning

Revisão

New

Observe que não usamos o modelo “Sequential” e sim o “Functional”, onde as camadas são funções aplicadas sobre as camadas anteriores...

Parte 3: Adicionar uma camada NN Densa com 10 perceptrons (ativação relu)

Parte 4: Adicionar uma camada NN com dois perceptrons (ativação softmax)

Transfer Learning

Revisão

Parte 5: Na compilação, mudar o loss para “binary_crossentropy”

Parte 6: Partindo das pastas de positivos e negativos, criar Xtn e ytn para o treinamento.

Parte 7:treinar com 100 épocas e depois 200. Observar o ajuste para Xtn[0]=O e Xtn[12]=X

Transfer Learning

Revisão

Houve treinamento de 12.821 parâmetros (treináveis, das novas camadas). Outros (2.257.984) vieram do treinamento original da MobileNet V2.

Observe que há apenas 10 positivos e 10 negativos no treinamento. Qual será o poder de um treinamento tão restrito que partiu da “poderosa” MobileNet?

Vamos testar com novas amostras (pasta oxteste)

Que tal aumentar esse dataset de testes?

Transfer Learning

Revisão

Vamos testar a acurácia de um modelo de CNN para o dataset O-X **sem** o Transfer Learning.

cnn_ox.ipynb

Os resultados **não** permitem uma boa comparação entre o modelo transfer e o modelo “puro cnn”, já que o treinamento e a validação foram feitos com dataset muito reduzido. A ideia foi somente mostrar os dois caminhos e **observar os resultados**.

Alguém se habilita a aumentar o dataset?

Sequence Learning

Conteúdo

Redes neurais para aprender sequências (séries temporais, textos etc.)

Redes Recorrentes RNN

Conteúdo

Antes de falar de RNN, vamos falar de recorrência.

Um exemplo: A formula de valor corrigido de um mês (m) para outro é:

$v(m)=v(m-1)*(1+i)$ **(1)**

Isso é uma recorrência, por que a definição do valor no mês “m” refere-se ao valor no tempo anterior, e é a mesma para todos os meses. Vamos ver um algoritmo recursivo em Python...

recursivo.ipynb

Para número finito de meses de correção, podemos apresentar a fórmula (1) de uma forma “unfolded”:

$v(3)=v(2)*(1+i)$

$v(2)=v(1)*(1+i)$

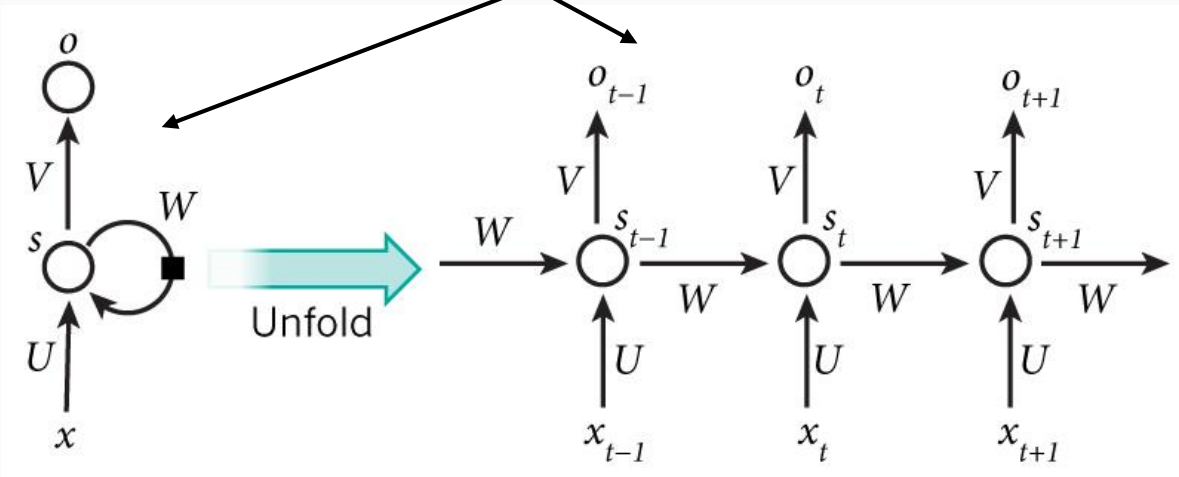
$v(1)=v(0)*(1+i)$

$v(0)=v$

Redes Recorrentes RNN

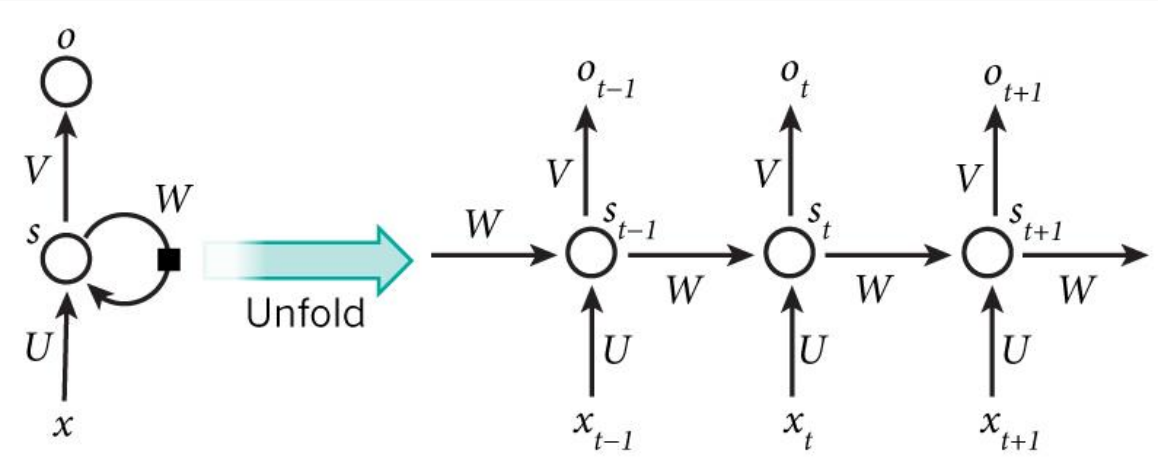
Conteúdo

Da mesma forma que podemos ter uma versão “unfolded” para expressões com recorrência, grafos computacionais com recorrência também podem ter uma forma compacta e uma “desempacotada”.



Redes Recorrentes RNN

Conteúdo



Um arranjo possível de RNN é o da figura acima

Há inputs x nos tempos t-1,t e t+1.

A “memória” (hidden state s) em t é dada por

$s_t=f(x_tU+Ws_{t-1})$

O output em t é $o_t=Vs_t$

W,U e V são respectivamente os pesos associados à memória, ao input e ao output.

Redes Recorrentes RNN

Podemos dizer que as redes neurais recorrentes são as que possuem algum arco de recorrência em seu grafo.

Assim, as redes feed forward aproximam funções “não recorrentes” e as RNN aproximam funções recorrentes...

Redes Recorrentes RNN

Conteúdo

Mais embasamento...

Ler até o segundo parágrafo da página 368.

Responder:

Qual é a vantagem do “sharing parameters”, algo que já vimos nas CNN.

Qual é a diferença entre fazer uma convolução em 1D e uma rede recorrente?

https://www.deeplearningbook.org/contents/rnn.html

Redes Recorrentes RNN

Conteúdo

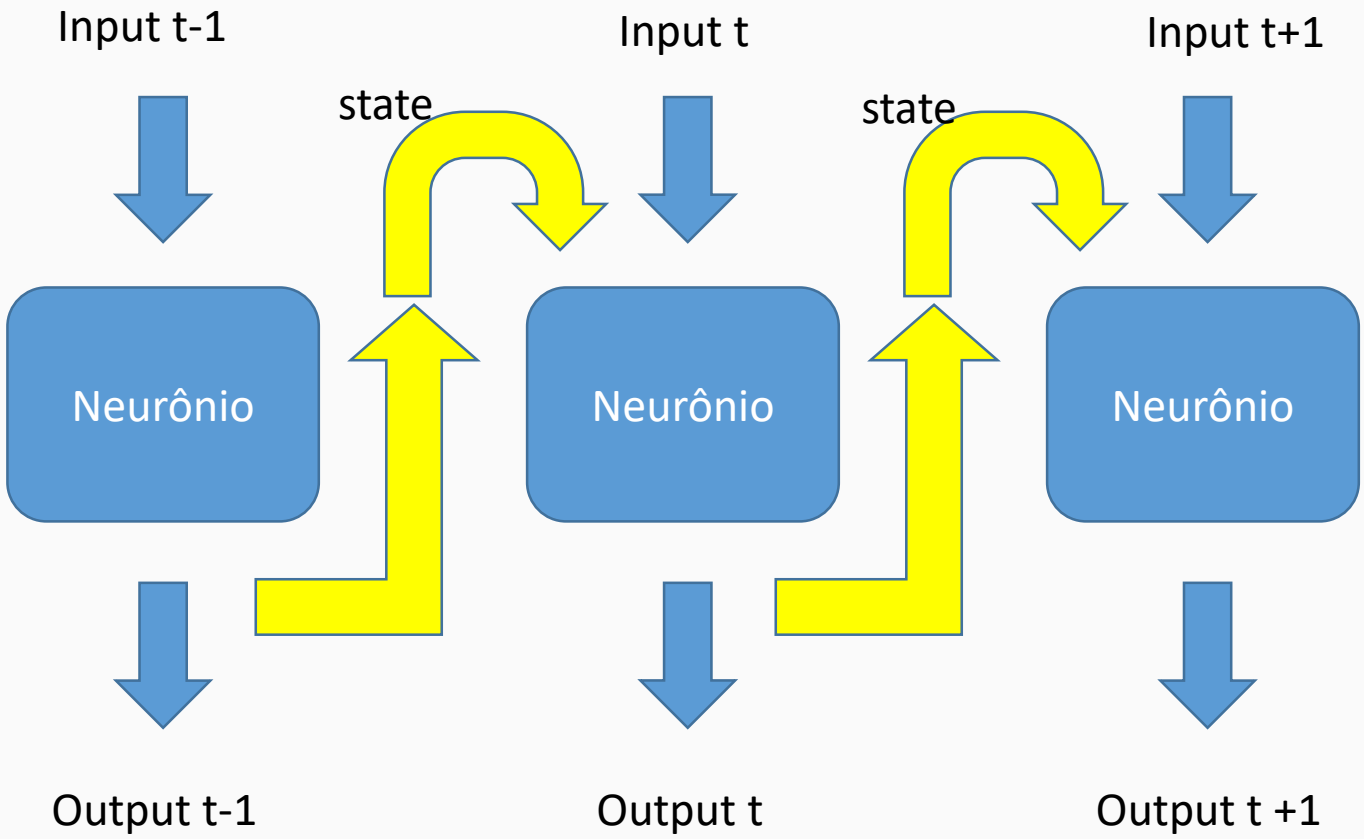
Por que precisamos das redes neurais recorrentes para processar time series (na verdade, sequências em geral) se já temos as redes feed forward?

As redes neurais recorrentes têm conexões com loops, adicionando feedback e memória às redes ao longo do tempo. Essa memória permite que esse tipo de rede aprenda e generalize através de seqüências de entradas, em vez de padrões individuais, como acontrece com as MLPs.

Redes neurais recorrentes são tipicamente usadas no processamento de sequências (textos, séries temporais etc.)

Recurrent Neural Networks (RNN)

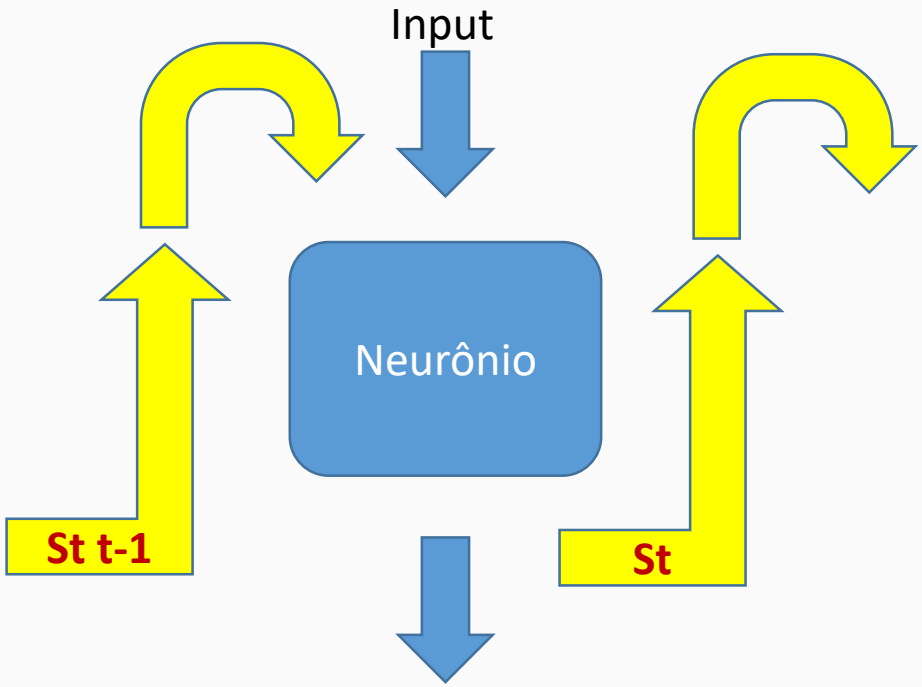
Mais um exemplo de RNN (visão “unfolded”) ao longo do tempo



Esse arranjo pode ser implementado com SimpleRNN com Keras

Recurrent Neural Networks (RNN)

Vamos implementar uma RNN que “adivinhará” o próximo número em sequência senos(x), com x variando de pi/10 em pi/10
Utilizaremos como ativação da célula, a função linear (identidade), g(x)=x.



Output= St=g(input*Wi+St_{t-1}*Ws+bias)

Partindo de **senoid_rnn.ipynb :**

1- Analisar código
2- Obtenha os pesos (model.get_weights())
3-Teste a predição de output com entrada [sen(0.31)] e [[sen(0.31)],sen(0.62)]] obtenha os mesmos valores a partir dos pesos

Partindo de **stock_rnn.ipynb :**

1- Analisar código (amostras com tamanho 10 e 4 células...houve necessidade de uma camada Dense(1). Por que?
2-Explique model.summary() e model.get_weight
3-Faça validação com split train test
4-Tente melhorar o Modelo

Cursos com Alta Performance de Aprendizado